Joan McCarthy
12744729

COMP41100 Exploring Programming with Ruby
Practical 9

Question 1

The methods (index, change and quote) each query the database for the last record in the database. This is not a very reliable method of querying for the required record as it may return the wrong entry when the entry is not stored last in the database. An alternative approach is to query the database by the id of the record that is being created. The id is returned as a parameter ech time there is a link to another page. This allows the next method to query the database for that record by its id. This modification has been made to the methods index, change, show and quote. An update method was also added to the controller in order to just update the salary and loan for the record instead of creating a new record with the new salary and loan values.

Other modifications included making fields in a form required so there wouldn't be any blank entries for a record in the database. Also in the name method, an entry is only created if the first name is not nil, this prevents empty records being entered into the database when page other/main is loaded.

Question 2

The rails app created is a simple blog. The root of the application displays the homepage. The homepage consists of a list of the blogs currently posted on the application. The user can add a new blog post by clicking on the '+' icon in the navigation bar. The user is then directed to the *post_new* page where a new post can be published to the blog. Here the user fills in the title, author and text content of the blog to be published. Once the user clicks the "Publish Post" button, the post details are saved to the database via the *post_new* method in the controller of the application. The user is then redirected to the homepage which now includes the new blog post.

A user can click on a blog post to view it in detail. This is achieved via the *post_detail* method within the *mysite* controller. From here a user can edit the post selected by following the "edit" link. The user can then update the title, author and text content of the selected blog post. A user also has an option to delete the post completely from the blog by clicking on the "delete" link within the *post_detail page*.

The navigation bar is part of the application.html.erb layout within the views therefore appears as a base for all pages of the application. This allows users to return to the homepage or ad a new post from any point of the application by just clicking on the links in the navigation bar. The navigation bar was created using bootstrap. Bootstrap was also used for table styling and displaying success messages on new post entry and post update or deletion.

Question 3

Other than clicking on links to other pages the user can type in the route of the required page into the address bar of the browser. This of course requires actually knowing the route of the required page. If an incorrect route is entered into the address bar while using the app on the rails server an error page will appear warning that there was no route matching that entered into the address bar. A table of the available routes is also displayed on this page.